Presented at the 2004 AIAA Guidance, Navigation, and Control Conference, August 16-19, Providence, RI.

Nonlinear Discrete-time Design Methods for Missile Flight Control Systems

P. K. Menon*, G. D. Sweriduk† and S. S. Vaddi‡
Optimal Synthesis Inc., Palo Alto, CA 94303

and

E. J. Ohlmeyer§
Naval Surface Warfare Center, Dahlgren, VA 22448

Discrete-time designs of flight control systems are required for implementation on missile-borne computers. While extensive literature is available on linear discrete-time control system design methods, nonlinear discrete-time control system design techniques have not been discussed to the same degree. This paper presents three nonlinear, discrete-time control system design methods. These are the discrete-time feedback linearization method, discrete-time state-dependent Riccatti equation method, and the discrete-time recursive backstepping technique. Nonlinear missile autopilot design for a conventional missile and an integrated guidance-control system design for a moving mass actuated missile are given as illustrative examples.

I. Introduction

Design methods for discrete-time linear control systems have reached an advanced level of maturity[1, 2]. However, the direct design of nonlinear discrete-time control systems remains to be fully developed. Although textbooks are available on nonlinear control system design[3 - 5], literature on discrete-time nonlinear control system design is rather sparse. From an applications point-of-view, discrete-time designs are important because most controllers are implemented using digital computers. Design techniques of interest in this paper are those that permit the synthesis of discrete-time controllers for continuous-time nonlinear dynamic systems.

The present work is motivated by the need to implement nonlinear control system designs synthesized using computer-aided design techniques[6, 7] onboard missiles. Three different discrete-time control system design techniques have been investigated in the present research. All of them are discrete-time analogs of continuous-time nonlinear system design techniques discussed in the literature. The first approach is the discrete-time version of the state-dependent Riccati equation (SDRE) technique discussed in References 8 and 9. The second design technique is a discrete-time version of the recursive backstepping[10] methodology, and employs discretized system dynamics. The third technique is the discrete-time version of the feedback linearization design approach. In this last technique, the system dynamics is first transformed into a linear, time-invariant form through the definition of state variable feedback. The transformed model is then converted into discrete-time form by defining sample-holds at the input and the outputs. The discretized is linear model is then used for control system design.

All three techniques have been employed for the design of missile flight control systems. Section II will present each of the design techniques in detail. Section III and IV will describe the missile control system design examples that illustrate the application of the design techniques. Numerical simulation results will also be presented, together with a discussion on real-time performance of the system on an off-the-shelf computer running the RT-Linux[11] operating system. Conclusions will be given in Section V.

---

* Chief Scientist, 868 San Antonio Road. Associate Fellow, AIAA.
† Research Scientist, 868 San Antonio Road. Member AIAA.
‡ Research Scientist, 868 San Antonio Road.
§ Senior Guidance and Control Engineer, Code G23. Associate Fellow, AIAA.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2004		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Nonlinear Discrete-time Design Methods for Missile Flight Control Systems	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Surface Warfare Center,Dahlgren,VA,22448**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**16**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

II. Discrete-time Nonlinear Control System Design Techniques

The three nonlinear discrete-time control design techniques discussed in this paper are (i) the State Dependent Riccatti Equation Method (ii) Recursive Backstepping Technique and (iii) the Feedback Linearization Method.

A block diagram of the closed loop systems incorporating a discrete-time controller is given in Figure 1. The dynamics of the system under control is given in the form of continuous-time nonlinear differential equations, while the controller computations occur at a specified sample rate. As a first-step, sample and hold circuits are introduced at the system inputs and outputs, which will be realized in practice using the analog-to-digital converters at the inputs and digital-to-analog converters at the output of the controller. Zeroth-order holds are assumed in the present research.

The continuous-time dynamics of the system is used as the basis for controller design. The first step in the design process is that of extracting a "design model" of the dynamic system. The design model should include all the important dynamic features of the system, excluding the dynamics of the sensors, and the actuators, if their dynamic responses are much faster than the system dynamics. Next, the variables in the system are separated into state and parameter vectors. The system parameter vector is made up of those constants and slowly varying variables that are not being controlled. For instance, in missiles that do not allow the direct control of engines, Mach number may be considered as a parameter. This separation also depends on the type of control system being designed. In missile autopilot design, the state vector may consist of angle of attack and pitch rate in the pitch axis, while the state vector in an integrated guidance-control system design may include lateral position state as an additional element. Other elements of the system parameter vector may consist of vehicle inertia properties, atmospheric density and the speed of sound. For the present research, the system dynamics is assumed to be of the form:

$$\dot{x} = f(x,p) + g(x,p)u \tag{1}$$

where x is the state vector, u is the control vector and p is a parameter vector. The parameter vector consists of a set of constants or slowly varying states that are not controlled by the closed-loop system. Note that the present development assumes that the controls appear linearly in the state equations. If this is not the case, a dynamic compensator at the input can be used to transform the system dynamics into this form[7, 9].

Next, the system performance requirements are generated. This may include the desired transient and steady state response characteristics. The controller sample rate can be chosen based on the expected closed-loop performance and the design model dynamic characteristics. The design techniques discussed in the following subsections can then be used to derive nonlinear controllers.

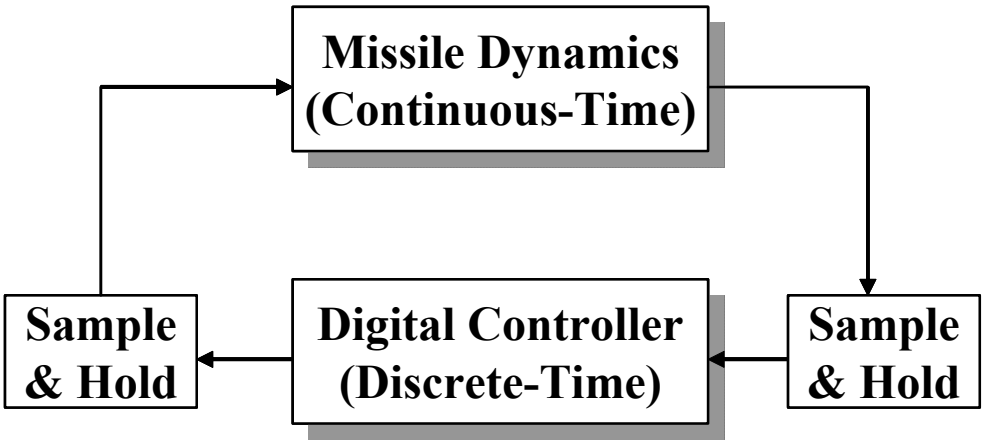

Figure 1. Discrete-time Closed Loop Control System

A. Discrete-Time State Dependent Riccatti Equation Method

The development in this section closely follows References 8 and 9. The first step in the SDRE method is the transformation of the system dynamics into the state dependent coefficient (SDC) form. The SDC form is an instantaneous parameterization of the original nonlinear system into the form:

$$\dot{x} = A(x)\,x + g(x)\,u. \tag{2}$$

Since *f(x)* is a vector and *A(x)* is a matrix, an infinite number of such realizations exist. However, from a numerical standpoint, only those parameterizations for which the pair [*A(x), g(x)*] is controllable at a given value of the state vector *x* should be used to set up the design procedure. Reference 7 has described a general numerical procedure for constructing the SDC parameterization from a dynamic model given in the form (1). Next, the SDC matrices [*A(x), g(x)*] are transformed into a discrete-time system using a sampling period of *T*. Since the value of the feedback state vector remains constant over a sample interval, define two new matrices:

$$A_d = e^{A(x[k])T}, \quad B_d = \left(\int_0^T e^{A(x[k])\tau}\,d\tau\right) g(x[k]) \tag{3}$$

The discretized form of the system dynamics in SDC form then becomes:

$$x(k+1) = A_d(k)\,x(k) + B_d(k)\,u(k) \tag{4}$$

The discretized SDC form of the system dynamics is then used to cast the control problem as an infinite-horizon nonlinear regulator problem with the goal of minimizing the cost function:

$$J = \frac{1}{2}\sum_{k=0}^{\infty}\left[x^t(k)\,Q(x[k])\,x(k) + u^t(k)\,R(x[k])\,u(k)\right] \tag{5}$$

where the *Q(x)* matrix is required to be positive semi-definite and *R(x)* must be positive definite, for all *x*. The solution to this problem can be obtained as[8, 9]:

$$u[k] = -K(x[k])\,x[k] \tag{6}$$

with the state-dependent feedback gain matrix computed as:

$$K(x[k]) = \left(B_d^T P(x[k])\,B_d + R(x[k])\right)^{-1} B_d^T P(x[k])\,A_d \tag{7}$$

The matrix *P* satisfies the state-dependent Riccati equation:

$$-P(x[k]) + A_d^T(x[k])\,P(x[k])\left[I + B_d R^{-1}(x[k])\,B_d^T P(x[k])\right]^{-1} A_d + Q(x[k]) = 0 \tag{8}$$

Note that every step in the above derivation can be numerically implemented. For the present research, algorithms have been built into the nonlinear system synthesis software[7] to directly work with a numerical simulation model. Using a perturbation vector, the SDC parameterization, model discretization and state-dependent gain computation are carried out by the design software.

The next two sub-sections will discuss recursive backstepping and feedback linearization techniques in the discrete-time setting. However, the following remarks on the difference between SDRE and the other two techniques are appropriate at this juncture. The only restriction the SDRE technique places on the design model is that the model be controllable. On the other hand, recursive backstepping and feedback linearization methods require the model to have a specific triangular structure.

B. Discrete-Time Recursive Backstepping Technique

Unlike the discrete-time SDRE method, or the discrete-time feedback linearization technique described in the next subsection, the discrete-time recursive backstepping approach requires the discretization of the nonlinear dynamic model as the first step. The continuous time nonlinear model can be converted into an approximate

discrete-time model using numerical integration techniques such as the Euler's method and the Adams-Bashforth method[12]. Moreover, the technique assumes that the system is given in a form:

$$\begin{aligned}
&x_1[k+1] = f_1(x_1[k]) + g_1(x_1[k])x_2[k] \\
&x_2[k+1] = f_2(x_1[k], x_2[k]) + g_2(x_1[k], x_2[k])x_3[k] \\
&\ldots\ldots\ldots\ldots = \ldots\ldots\ldots\ldots\ldots\ldots \\
&x_i[k+1] = f_i(x_1[k], x_2[k], \ldots x_i[k]) + g_i(x_1[k], x_2[k], \ldots x_i[k])x_{i+1}[k] \\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&x_n[k+1] = f_n(x_1[k], x_2[k], \ldots x_n[k]) + g_n(x_1[k], x_2[k], \ldots x_n[k])u[k]
\end{aligned} \tag{9}$$

It should be noted that the $f_i^{'}s$ and $g_i^{'}s$ in the above state equations could also be functions of a parameter vector *p*. In the case where *m* control variables are available, it is assumed that the system dynamics can be partitioned and arranged with respect to each of the control variables. The recursive backstepping technique proceeds by recursively constructing a Lyapunov function with respect to each control variable, in a recursive manner. As a first step, a quadratic Lyapunov function is assumed for the first state as follows:

$$V_1[k] = \frac{1}{2}x_1^2[k] \tag{10}$$

Since the discrete-time Lyapunov stability analysis requires the Lyapunov function to be a contraction mapping,

$$V_1[k+1] = \eta_1^2 V_1[k], \ \ where\ \eta_1 \in [0\ 1] \tag{11}$$

This implies

$$(f_1[k] + g_1[k]x_2[k])^2 = \eta_1^2 x_1^2[k] \tag{12}$$

Treating the second state x_2 as a control-like variable for x_1, the desired value of x_2 can be computed as:

$$x_{2c}[k] = \frac{-f_1[k] \pm \eta_1 x_1[k]}{g_1[k]} \tag{13}$$

The above equation offers two choices for the reference trajectory. Although both choices lead to a stable system, the following solution is adopted in this work

$$x_{2c}[k] = \frac{-f_1[k] + \eta_1 x_1[k]}{g_1[k]} \tag{14}$$

For an equivalent linear dynamic system, the above solution results in a closed-loop pole location on the positive real axis. The other choice would have resulted in a closed pole location on the negative real axis. This is not desirable in discrete-time dynamic systems, due to the fact it places the system operation close to the Nyquist rate[1]. It should be noted that the choice $\eta \in [0\ 1]$, ensures the closed loop pole of the equivalent linear system lies within the unit circle.

The Lyapunov function at the second stage of the backstepping process is chosen as:

$$V_2[k] = \frac{1}{2}(x_2[k] - x_{2c}[k])^2 \tag{15}$$

Invoking the stability analysis and treating the third state as a control-like variable, an expression for $x_{3c}[k]$ can be derived as follows:

$$V_2[k+1] = \eta_2^2 V_2[k], \ where\ \eta_2 \in [0\ 1] \tag{16}$$

From which,

$$(f_2[k] + g_2[k]x_{3c}[k] - x_{2c}[k+1])^2 = \eta_2^2(x_2^2[k] - x_{2c}^2[k]) \tag{17}$$

Consequently,

$$x_{3c}[k] = \frac{-f_2[k] + x_{2c}[k+1] + \eta_2(x_2[k] - x_{2c}[k])}{g_2[k]} \tag{18}$$

As in the previous stage of the recursive backstepping process, the solution corresponding to the positive real axis location of the closed loop pole has been chosen. Note that the above solution requires $x_{2c}[k+1]$ at the k^{th} time step. This can be obtained by propagating Equation (14).

$$x_{2c}[k+1] = \frac{-f_1[k+1] + \eta_1 x_1[k+1]}{g_1[k+1]} \tag{19}$$

Since the right hand side depends only on $x_1[k]$ and $x_2[k]$, it can be evaluate at the k^{th} time step. It should be noted that the overall Lyapunov function for the first and second states is $V_1[k] + V_2[k]$. It can be shown that for the above choice of η_1, η_2 the overall Lyapunov function also is a contraction mapping[10].

Proceeding along similar lines, the reference for the i^{th} state in the control chain can be written as:

$$x_{ic}[k] = \frac{-f_{i-1}[k] + x_{(i-1)c}[k+1] + \eta_{i-1}(x_{i-1}[k] - x_{(i-1)c}[k])}{g_{i-1}[k]} \tag{20}$$

And the control at the k^{th} time step can be obtained as:

$$u[k] = \frac{-f_n[k] + x_{nc}[k+1] + \eta_n(x_n[k] - x_{nc}[k])}{g_n[k]} \tag{21}$$

The overall Lyapunov function for the recursive backstepping-based closed-loop system is:

$$V[k] = V_1[k] + V_2[k] + ...V_n[k] \tag{22}$$

The first term in the Lyapunov function is a quadratic term containing just the leading state, and the other terms are:

$$V_i[k] = \frac{1}{2}(x_i[k] - x_{ic}[k])^2, \quad i = 2,3....n. \tag{23}$$

It may be observed that the recursive backstepping method automatically constructs a Lyapunov function for the dynamic system during the design process. This can be used not only for control system design, but also for stability and robustness analysis of the closed-loop system.

C. Discrete-Time Feedback Linearization Method

As in the recursive backstepping technique, the feedback linearization technique[3 - 7, 13] assumes that the system dynamics can be partitioned with respect to each control variable, and arranged in a triangular structure, as follows:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1) + g_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2) + g_2(x_1, x_2, x_3) \\ &\vdots \\ \dot{x}_n &= f_n(x_1, x_2, \cdots, x_n) + g_n(x_1, x_2, \cdots, x_n)u \end{aligned} \tag{24}$$

The technique then uses repeated differentiation of the leading state equation and substitutions to transform the model into *Brunovsky* canonical form[14]. In this form, the system dynamics appear as chains of integrators with all the system nonlinearities moved to the input:

$$x_1^{...n} = f(x_1, x_2, ...x_n) + g(x_1, x_2, ...x_n)u \tag{25}$$

Such chains of integrators can be constructed for each of the control variables. The right hand side of equation (25) is then replaced with a pseudo control variable v to yield the transformed system:

$$x_1^{...n} = v = f(x_1, x_2, ...x_n) + g(x_1, x_2, ...x_n)u \tag{26}$$

This transformation makes the dynamic system linear and time-invariant with respect to the pseudo-control variables. Linear control techniques can then be used to design controllers for the transformed model. For the case of a single control input, the transformed model can be written in state-space form as shown below:

$$\dot{z} = Az + Bv \tag{27}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{28}$$

Next, sample and hold circuits are introduced at every control input and state to obtain a discrete-time, linear dynamic system[1] as:

$$z(k+1) = A_d z(k) + B_d v(k) \tag{29}$$

Discrete-time linear control techniques such as pole-placement and linear quadratic regulator theory[15] can be used to design feedback control laws for the system. For instance, an optimal discrete-time control law can be designed as:

$$v(k) = G(k) z(k) \tag{30}$$

where, the gain matrix *G(k)* is obtained from the solution of a discrete-time Riccati equation[15]. The actual control variables can then be recovered from the pseudo-control variables using the relationship:

$$u[k] = g(x[k])^{-1}\left[v(z[k]) - f(x[k])\right] \tag{31}$$

If the system nonlinearities are known reasonably well and the sampling rate chosen appropriately, the resulting closed-loop system will have dynamic properties close to the discrete-time linear system. An interesting feature of the feedback linearization approach is that once the system dynamics is transformed into linear, time-invariant form, any control system design technique can be applied to derive the controller. For instance, the H_∞ control technique[16] or the sliding mode control[3] technique can be applied to the transformed control problem.

The next two sections will illustrate the application of these three discrete-time nonlinear control system design techniques to two missile flight control system design examples.

III. Missile Longitudinal Autopilot Design

The first design example considers an angle of attack regulation problem of a hypothetical tail-controlled missile model from Reference 17. The equations of motion given in the following consists of angle of attack α, pitch rate q, Mach number M and flight path angle γ as the state variables and employs pitch fin deflection δ as the control input.

$$\begin{aligned}\dot{\alpha} &= 0.4008\, M\alpha^3 \cos(\alpha) - 0.6419\, M|\alpha|\alpha\cos(\alpha) \\ &\quad - 0.2010\, M\left(2 - \frac{M}{3}\right)\alpha\cos(\alpha) - 0.0403\, M\cos(\alpha)\delta \\ &\quad + 0.0311\frac{\cos(\gamma)}{M} + q\end{aligned} \tag{32}$$

$$\begin{aligned}\dot{q} &= 49.82\, M^2\alpha^3 - 78.86\, M^2|\alpha|\alpha + 3.60\, M^2\left(-7 - \frac{8M}{3}\right)\alpha \\ &\quad - 14.54\, M^2\delta - 2.12 M^2 q\end{aligned} \tag{33}$$

$$\begin{aligned}\dot{M} &= 0.4008\, M^2\alpha^3 \sin(\alpha) - 0.6419\, M^2|\alpha|\alpha\sin(\alpha) \\ &\quad - 0.2010\, M^2\left(2 - \frac{M}{3}\right)\alpha\sin(\alpha) - 0.0062\, M^2 \\ &\quad - 0.0403\, M^2\sin(\alpha)\delta - 0.0311\sin(\gamma)\end{aligned} \tag{34}$$

$$\dot{\gamma} = -0.4008\, M\alpha^3 \cos(\alpha) + 0.6419\, M|\alpha|\alpha\cos(\alpha) + 0.2010\, M\left(2 - \frac{M}{3}\right)\alpha\cos(\alpha) + 0.0403\, M\cos(\alpha)\delta - 0.0311\,\frac{\cos(\gamma)}{M} \tag{35}$$

The control objective is to design an autopilot that stabilizes the missile airframe and tracks a commanded angle of attack or normal acceleration. As is customary in missile flight control system design[18], only the short-period dynamics of the missile consisting of the angle of attack α and pitch rate q dynamics are used in the autopilot design. Mach number M and flight path angle γ are treated as parameters in the short period model that are available as measurements.

Moreover, eventhough the fin deflection appears explicitly on the right hand side of the angle of attack equation its effect is ignored in recursive backstepping method and the feedback linearization approach. It is well known[18] that the effect of fin deflection on the angle of attack equation causes the nonminimum phase dynamic behavior in tail controlled missiles. Finally, since missiles are capable of generating several g's of lateral acceleration, the acceleration due to gravity is neglected.

The three different nonlinear discrete-time controller designs based on the above assumptions have been developed for the longitudinal missile dynamics model. The results given in this section will illustrate the regulation of the angle of attack about zero. The initial conditions for the closed loop simulation are chosen as $\alpha(0) = 0.1, q(0) = 0.01, \gamma(0) = -0.01$ and $M(0) = 2.5$, and a sampling period of 0.05 seconds is used in all the control system designs. The state weighting matrix Q and the control weighting R for the SDRE design method are chosen as constants.

$$Q = \begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix} \quad R = 100 \tag{36}$$

Figures 2 and 3 show the closed loop simulation results using SDRE method:

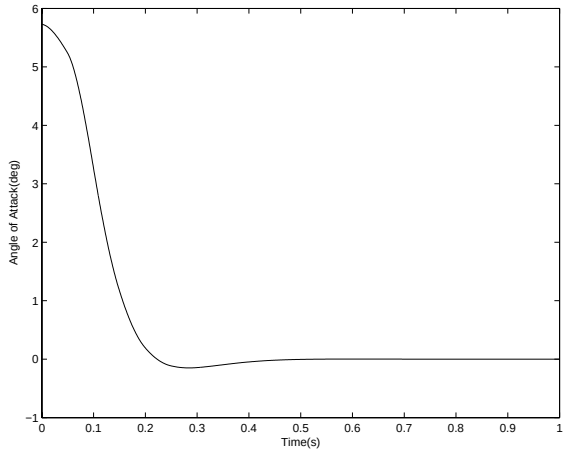

Figure 2. Angle of Attack History for the Discrete-Time SDRE Controller

The discrete-time recursive backstepping controller is designed is obtained by discretizing the continuous time model using Euler's method, with the same sampling period as the SDRE method. The convergence rates used are $\eta_1 = 0.6$ for angle of attack and $\eta_2 = 0.3$ for pitch rate. Figures 4 and 5 show the closed loop simulation results using discrete-time recursive backstepping autopilot.

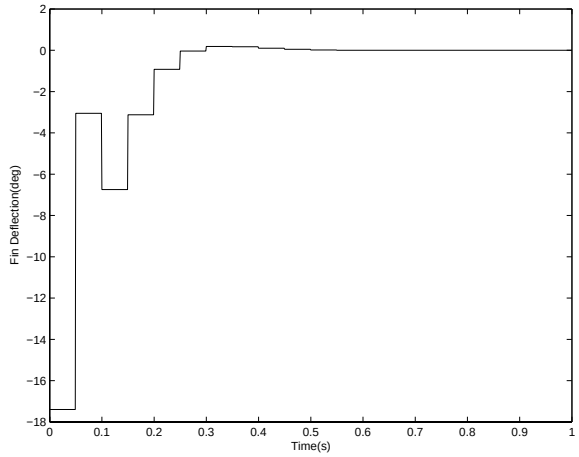

Figure 3. Fin Deflection History for the Discrete-Time SDRE Controller

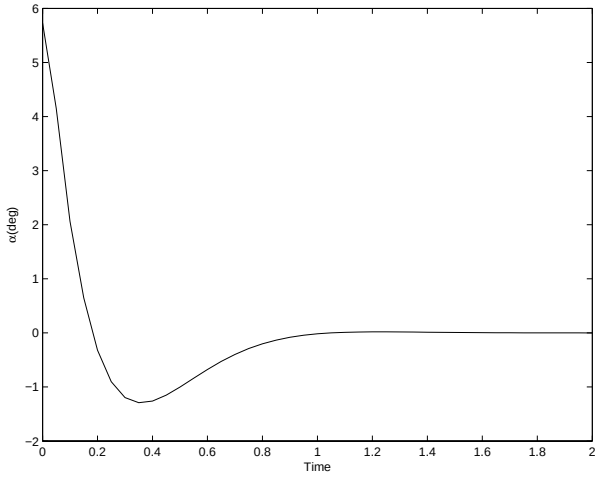

Figure 4. Angle of Attack History for the Discrete-Time Recursive Backstepping Controller

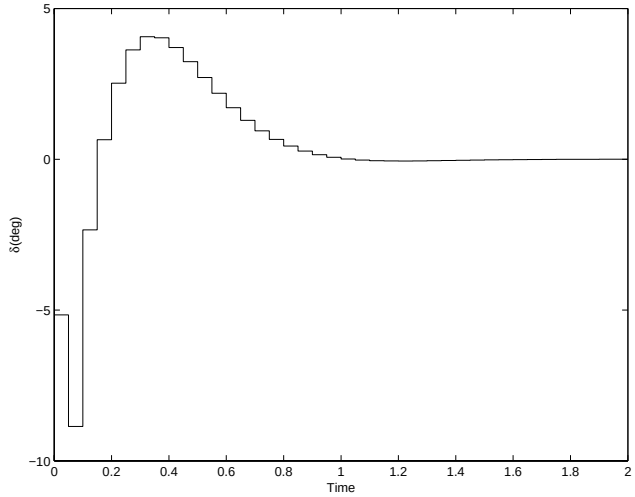

Figure 5. Fin Deflection History for the Discrete-Time Recursive Backstepping Controller

The behavior of the discrete-time Lyapunov function for the closed loop system with the recursive backstepping controller is given in Figure 6. The constant convergence rate of the Lyapunov function can be observed.

The third autopilot design is based on the discrete-time feedback linearization methodology. For the present research, a pole placement controller was chosen for controlling the linear dynamics resulting after performing the feedback linearization. The discrete-time closed-loop pole locations are chosen as: [*3.7486e-01 + i3.2032e-01 3.7486e-01 - i3.2032e-01*]. The sampling period of 0.05 seconds is retained for this technique as well. Figures 7 and 8 show the closed loop simulation results for the discrete-time feedback linearized autopilot.

The autopilot performance in all three cases is qualitatively similar. It is important to note that these designs could be improved considerably by iterating on the design parameters.

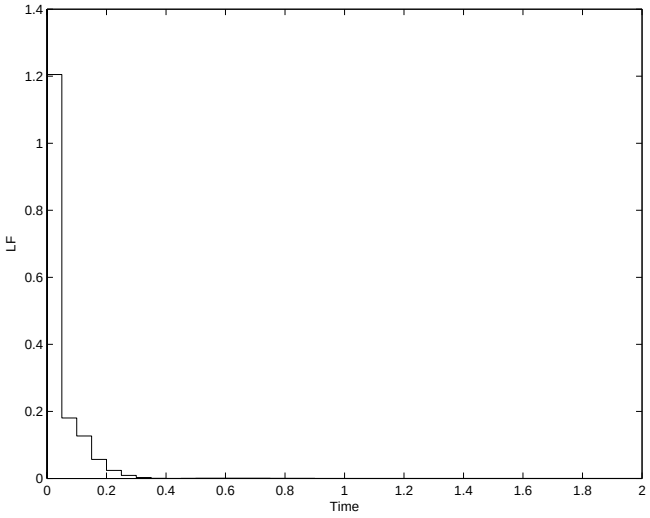

Figure 6. Time History of the Lyapunov Function for the Closed-Loop System with Discrete-Time Recursive Back Stepping Controller

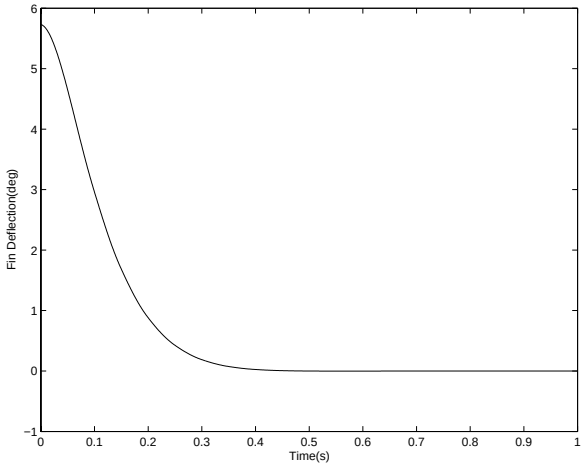

Figure 7. Angle of Attack History for the Discrete-Time Feedback Linearized Autopilot

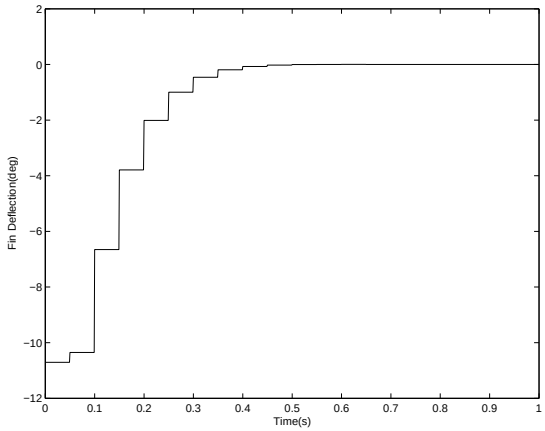

Figure 8. Fin Deflection History for the Discrete-Time Feedback Linearized Autopilot

IV. Integrated Flight Control System Design for a Moving-Mass Actuated Interceptor

Recently, internal mass movement has been proposed as a control methodology for a kinetic warhead (KW) in atmospheric and exo-atmospheric engagements[19]. As shown in Figure 9, the moving-masses positioned by servos inside the vehicle changes the location of its center of mass relative to the external forces to generate the desired control moments. The moving-mass control concept works equally well in space when the KW is thrusting, or in the atmosphere, when the vehicle experiences aerodynamic forces. An advantage of this actuation technology is that it can be employed in kinetic warheads that have both atmospheric and exo- atmospheric interception capabilities.

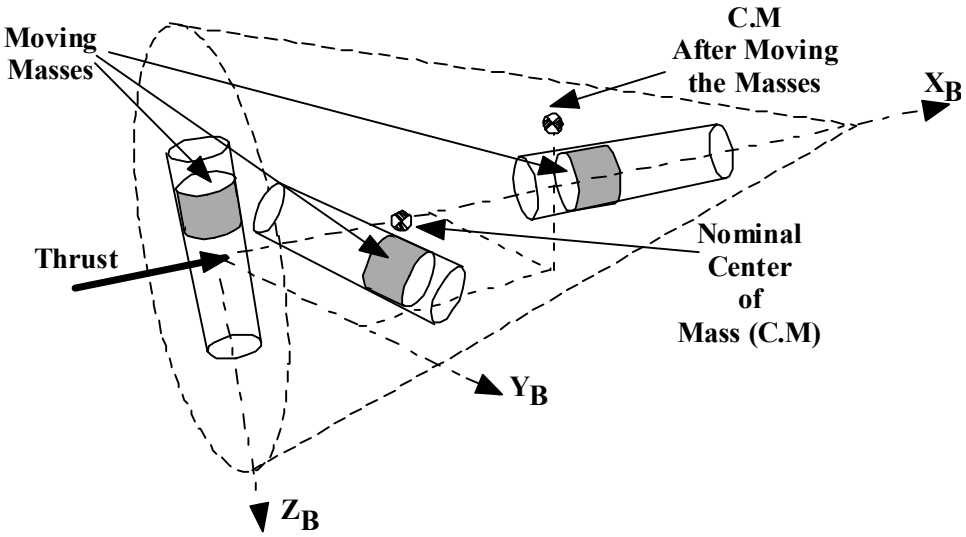

Figure 9. Moving-Mass Control Concept

As discussed in Reference 19, the moving mass actuator controlled kinetic warhead is a high-order coupled nonlinear dynamic system. The complexity of the model along with the varying aerodynamic characteristics of the vehicle makes it essential to use computer-aided nonlinear control system design techniques. The benefits of employing computer-aided nonlinear control system design techniques for design of continuous time control systems for the kinetic warhead target's interception were demonstrated in Reference 19. This section will illustrate the use of discrete-time feedback linearization method in conjunction with discrete-time pole placement methodology for controlling the moving-mass KW.

As in Reference 19, the present kinetic warhead (KW) includes two masses that can be moved along the body-frame y and z directions. The task of interception is achieved by driving the line-of-sight rates $\dot{\lambda}_z, \dot{\lambda}_y$ between the KW and the target to zero. The component of the relative motion vector along longitudinal axis of the KW is uncontrollable.

The control influence chains remain the same as they were in Reference 19. However, the control commands are position commands (δ_{zc} and δ_{yc}) to the actuators positioning the masses instead of the force commands. The moving-mass actuator proportional + derivative servos have a bandwidth of about 40 Hz.

$$\begin{aligned} \delta_{zc} &\to \dot{\delta}_z \to \delta_z \to q \to w \to \dot{\lambda}_z \\ \delta_{yc} &\to \dot{\delta}_y \to \delta_y \to r \to v \to \dot{\lambda}_y \end{aligned} \tag{37}$$

The closed loop pole locations are chosen as {0.6005 0.6065 0.8607 0.9048 0.9900} for both the pitch and yaw channels. The roll channel does not incorporate closed-loop control. A sample frequency of 100 Hz is used in the design.

This engagement scenario considered in this section has the warhead and target initially at an altitude of 45,000 ft, and 50,000 ft. apart. Both vehicles have 15-degree flight path angles and are on reciprocal headings, and both have an initial velocity of 6000 ft/sec. In addition, the target has an initial offset of 1000 ft. in the east direction. The performance of the kinetic warhead is shown in Figures 10 through 15. Comparison with the results in Reference 19 show that the discrete-time feedback linearized controller performance is close to the continuous-time case. The miss-distance for this maneuver was 0.00023412(ft).

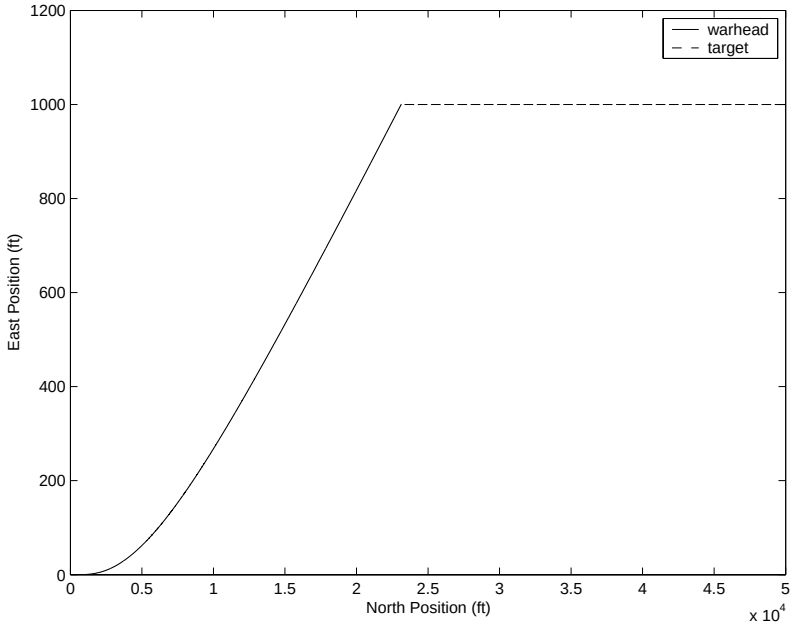

Figure 10. KW and Target Trajectories in the Horizontal Plane

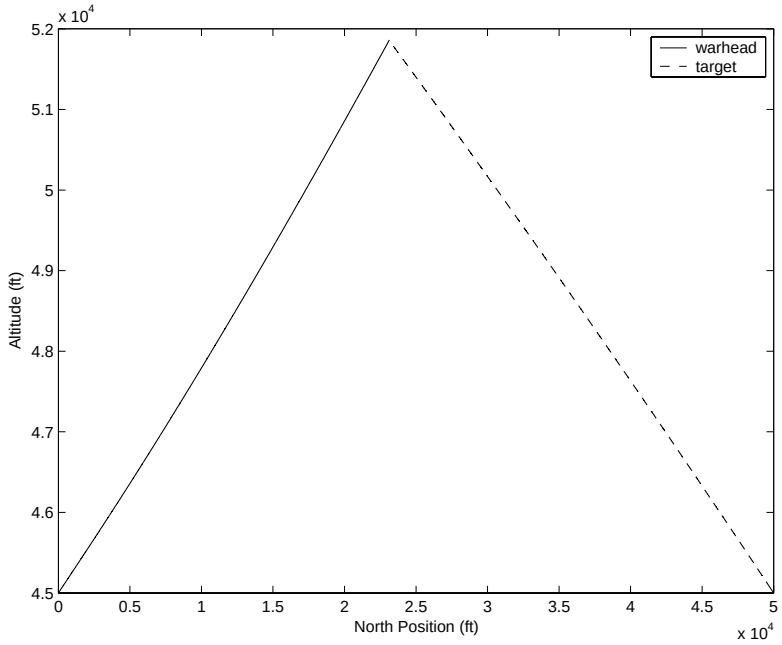

Figure 11. KW and Target Trajectories in the Vertical Plane

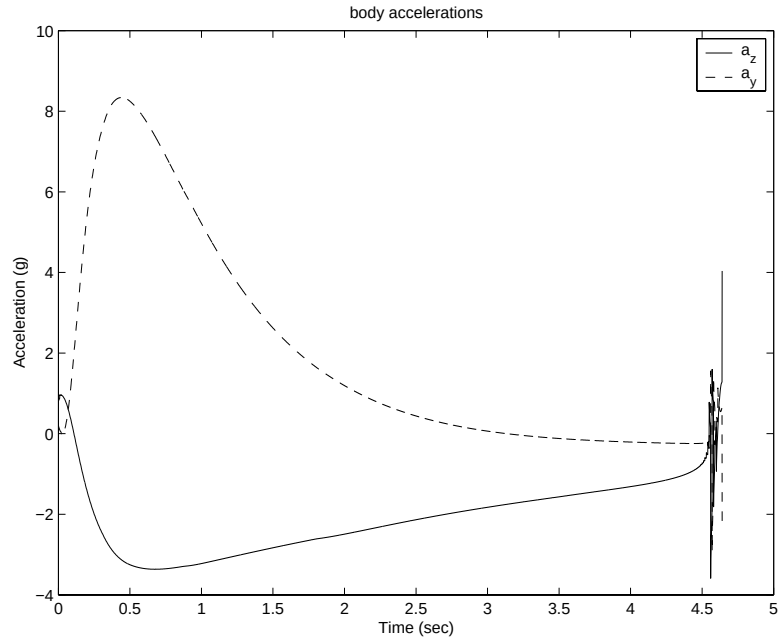

Figure 12. Lateral Acceleration Components of the KW during Interception

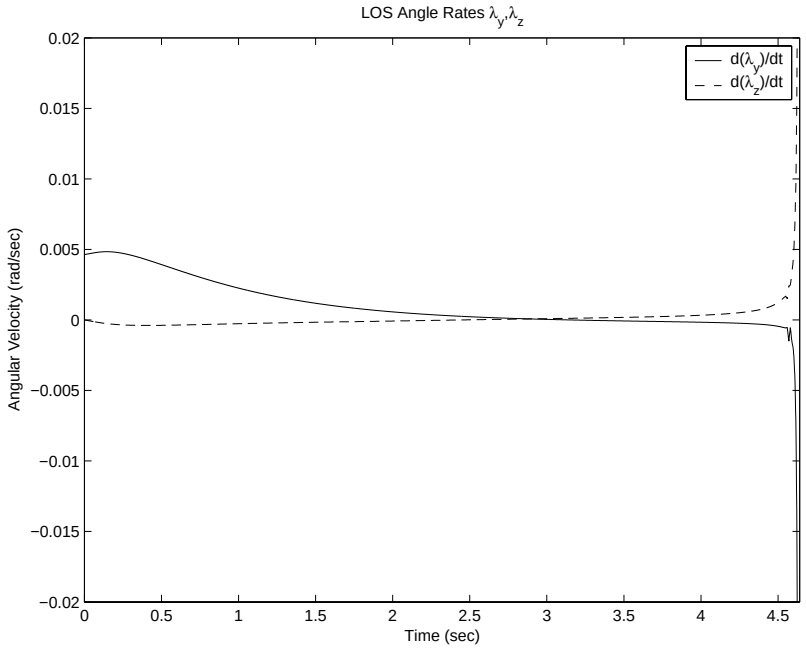

Figure 13. KW Line-of-Sight Rates During the Interception

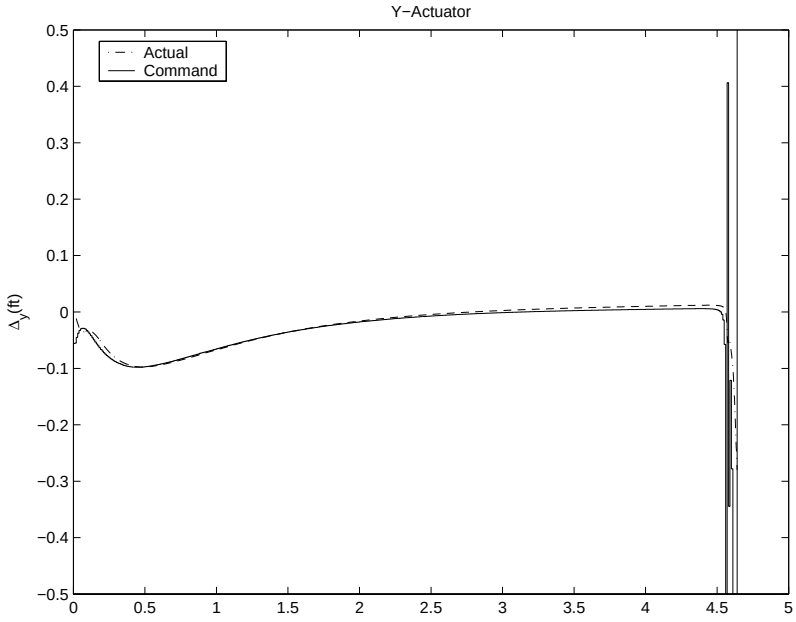

Figure 14. Actual and Commanded Positions of the Moving Mass in the Yaw Axis

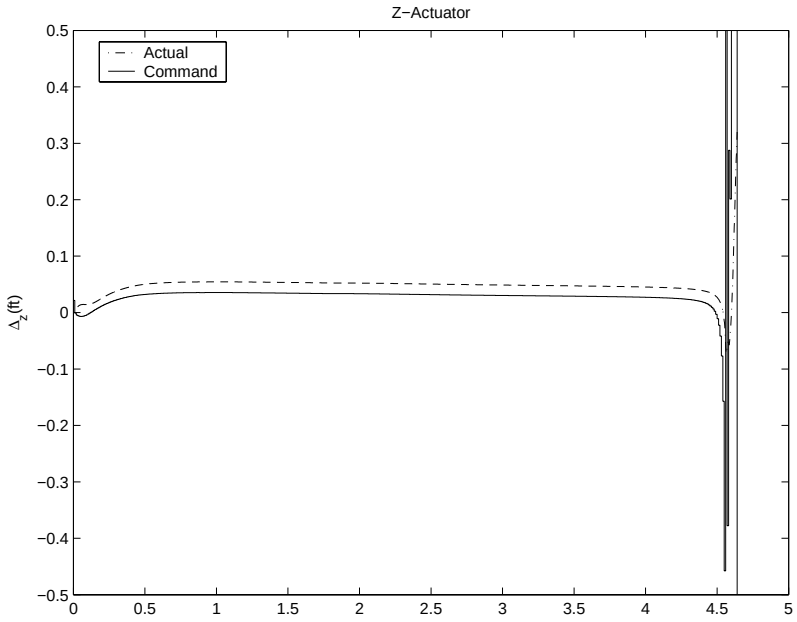

Figure 15. Actual and Commanded Positions of the Moving Mass in the Pitch Axis

As with classical proportional navigation guidance law, the presented integrated guidance-control system design generates large amplitude maneuvers towards the end of the maneuver.

Real-Time Evaluation of the Nonlinear Integrated Guidance-Control System

The discrete-time nonlinear control law described in the foregoing has been evaluated in a real-time computing environment. The computer-aided nonlinear control system synthesis software currently has the capability to automatically generate C code from the user-specified design parameters. Freely available LAPACK[20] linear algebra routines are used in this process.

The guidance-control law was implemented in a Pentium III, 800 Mhz computer running the RTLinuxFree[11] operating system. This real-time operating system is community-supported and permits free open source distribution of the operating system under the General Public License (GPL). Several runs have been made, and the results indicate that the discrete-time nonlinear guidance-control law can be safely operated at sample intervals of about 10ms. Moreover, further speed improvements appear feasible by reorganizing the C-code modules. This experiment indicates that the integrated guidance-control law discussed in this section can be implemented on state-of-the-art airborne computers.

V. Conclusion

This paper discussed three different discrete-time nonlinear flight control system design techniques. These were the discrete-time SDRE technique, discrete-time recursive backstepping technique, and the discrete-time feedback linearization approach. These techniques have been integrated with computer-aided nonlinear control system design software. The utility of these techniques was illustrated using two missile flight control system design examples. The feasibility of real-time implementation on a commercial off-of-the-shelf computer was investigated for one of the design examples. The results given here demonstrate that nonlinear discrete-time control system designs can be carried out in a manner similar to the continuous-time designs.

Acknowledgments

This research was supported under the MDA/Navy Contract No. N00178-03-C-3061.

References

[1]Ogata, K., *Discrete-Time Control Systems*, Prentice-Hall, Upper Saddle River, NJ, 1995.

[2]Franklin, G. F., and Powell, J. D., *Digital Control*, Addison-Wesley, Menlo Park, CA 1980.

[3]Slotine, J. E., and Li, W., *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[4]Isidori, A., *Nonlinear Control Systems*, Springer-Verlag, New York, NY, 1985.

[5]Marino, R., Tomei, P., *Control Design: Geometric, Adaptive and Robust*, Prentice-Hall, New York, NY, 1995.

[6]Menon P. K, Iragavarapu V. R, Sweriduk G and Ohlmeyer E. J, "Software Tools for Nonlinear Missile Autopilot Design," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 9-11, 1999, Portland, Oregon.

[7]Menon, P. K., Cheng, V. H. L., Lam, T., Crawford, L. S., Iragavarapu, V. R., and Sweriduk, G. D., *Nonlinear Synthesis Tools™ for Use with MATLAB®*, Optimal Synthesis Inc., 2004, Palo Alto, CA.

[8]Cloutier, J. R., "State-Dependent Riccati Equation Techniques: An Overview," *Proceedings of the American Control Conference*, Albuquerque, NM, June 4-6, 1997, pp. 932-936.

[9]Cloutier, J. R., D'Souza, C. N. and Mracek, C. P., "Nonlinear Regulation and Nonlinear H_∞ Control Via the State Dependent Riccati Equation Technique, Part 1: Theory, Part 2: Examples," *Proceedings of the International Conference on Nonlinear Problems in Aviation and Aerospace*, Daytona Beach, FL, May 1996.

[10]Kristi'c, M., Kanellakopoulos, I., and Kokotovi'c, P., Nonlinear and Adaptive Control Design, Wiley, New York, NY, 1995.

[11]http://www.fsmlabs.com/products/openrtlinux/

[12]Gerald, C. F., *Applied Numerical Analysis*, Addison-Wesley, Menlo Park, CA , 1978.

[13]Brockett, R. W., "Nonlinear Systems and Differential Geometry," *Proceedings of the IEEE*, Vol. 64, No. 1, Feb. 1976, pp. 61-72.

[14]Kailath, T., *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[15] Bryson, A. E., and Ho, Y. C, *Applied Optimal Control*, Hemisphere, New York, NY, 1975.

[16]Burl, J. B., *Linear Optimal Control: H_2 and H_∞ Methods*, Addison-Wesley, Menlo Park, CA, 1999.

[17]Mracek, C. P., and Cloutier, J. R., "Missile Longitudinal Autopilot Design Using the State Dependent Riccati Equation Method*," Proceedings of the International Conference on Nonlinear Problems in Aviation and Aerospace*, Daytona Beach, FL, May 1996.

[18]Blakelock, *Automatic Control of Aircraft and Missiles*, Wiley, New York, NY, 1965.

[19]Menon P. K., Sweriduk G., Ohlmeyer E. J., and Malyevac D. S. "Integrated Guidance and Control of Moving Mass Actuated Kinetic Warheads", *Journal of Guidance Control and Dynamics*, Vol. 27, No. 1, Jan-Feb 2004, pp. 118-126.

[20]Anderson, F., et al, *LAPACK User's Guide*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, August 1999. (http://www.netlib.org/lapack).